IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, April 2025
Impact Factor: 7.67

Smart Solar Tracker using Arduino Nano and LDR Sensors

Prof. Mane M. A., Sarthak Rajendra Dhamale, Ketan Rajendra Dongare, Someshwar Yuvraj Katkar, Rohan Rajendra Girigosavi, Om Ramesh Balshankar
Department of Electronics And Telecommunication
JSPM's Bhivrabai Sawant Polytechnic, Wagholi, Pune

Abstract: *As the world transitions toward cleaner and more sustainable energy sources, solar energy has emerged as one of the most promising alternatives to fossil fuels. However, the efficiency of solar panels is significantly influenced by their orientation with respect to the sun. Traditional static panels only perform optimally when the sun is directly overhead, resulting in suboptimal energy collection for the majority of the day. To overcome this limitation, solar tracking systems have been developed to follow the sun's movement and maximize energy absorption.*

This project focuses on designing and implementing a smart solar tracker using an Arduino Nano microcontroller and four digital Light Dependent Resistors (LDRs). The system dynamically adjusts the orientation of a solar panel to follow the sun throughout the day. Four LDRs are strategically positioned to detect sunlight intensity from different directions. Based on these readings, the Arduino controls a servo motor to rotate the panel towards the direction with the highest light intensity. The system ensures that the panel remains optimally aligned with the sun as it moves from east to west.

The project is cost-effective, simple to build, and suitable for both educational purposes and small-scale solar applications. By increasing the exposure of the solar panel to direct sunlight throughout the day, the tracker significantly enhances the overall efficiency of the solar power system. The smart solar tracker offers a practical solution to improving solar panel performance using basic electronic components and microcontroller programming....

Keywords: maximize energy absorption

I. INTRODUCTION

Solar tracking systems are designed to orient solar panels toward the sun to maximize energy capture. Conventional solar panels are fixed and only gather maximum energy during a short period when the sun is directly overhead. This project implements a simple, effective solar tracker using an Arduino Nano, LDR sensors, and a servo motor. The system automatically aligns the panel to the brightest light source, significantly increasing its efficiency.

II. LITERATURE SURVEY

Several studies and projects have explored solar tracking technologies to address the limitations of static solar panels. The literature reveals a wide range of designs, from simple mechanical trackers to complex microcontroller-based dual-axis systems.

Microcontroller-Based Trackers: Previous research shows the use of microcontrollers like PIC, ATmega, and Arduino boards for solar tracking applications. These controllers read sensor inputs (typically LDRs or photodiodes) and actuate motors to adjust panel positions. In particular, Arduino-based systems stand out due to their ease of programming, open-source community support, and wide availability.

LDR-Based Tracking: Multiple works have implemented LDRs as sunlight sensors, utilizing the variation in resistance with light intensity to determine the sun's position. Studies demonstrate that placing LDRs in a cross-formation enables accurate directional detection. Some researchers also experimented with shadow casting over LDRs for improved sensitivity.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, April 2025
Impact Factor: 7.67

Servo Motor Integration: Lightweight panels and DIY applications often use servo motors due to their simplicity and precise angle control. This makes them ideal for projects using microcontrollers with limited PWM outputs. Studies have also documented the ability of servo motors to respond to incremental light changes for smooth tracking.

Performance Improvement: It has been documented that single-axis trackers can improve energy absorption by 25-35%, while dual-axis trackers may increase efficiency by up to 40%. For small-scale and educational setups, single-axis Arduino-based trackers offer a good balance between complexity and performance.

Energy Efficiency and Cost Analysis: Researchers agree that automatic solar trackers provide a better return on investment over time compared to fixed systems. Although the initial setup may be slightly more expensive, the increase in power output compensates for it.

In summary, the literature highlights that using Arduino, LDRs, and servo motors is an effective and proven method for constructing a functional and efficient solar tracker. This project builds upon previous work and adds simplicity and practical usability suitable for students, hobbyists, and small installations.

III. COMPONENTS REQUIRED

Component	Quantity	Description
Arduino Nano	1	Microcontroller board for controlling system logic
LDR Sensors (Digital)	4	Detect light intensity in four directions
Servo Motor (SG90)	1	Adjusts the panel angle based on light direction
Breadboard	1	For easy prototyping and connection
Jumper Wires	As needed	To make necessary connections
Power Supply (5V)	1	To power Arduino and servo motor
USB Cable	1	For programming Arduino Nano

IV. CIRCUIT DIAGRAM

Connections:
LDR1 -> D2 (Digital Pin 2 of Arduino Nano)
LDR2 -> D4 (Digital Pin 4 of Arduino Nano)
LDR3 -> D5 (Digital Pin 5 of Arduino Nano)
LDR4 -> D6 (Digital Pin 6 of Arduino Nano)
Servo Signal -> D3 (Digital Pin 3 of Arduino Nano)
VCC of all modules -> 5V
GND of all modules -> GND

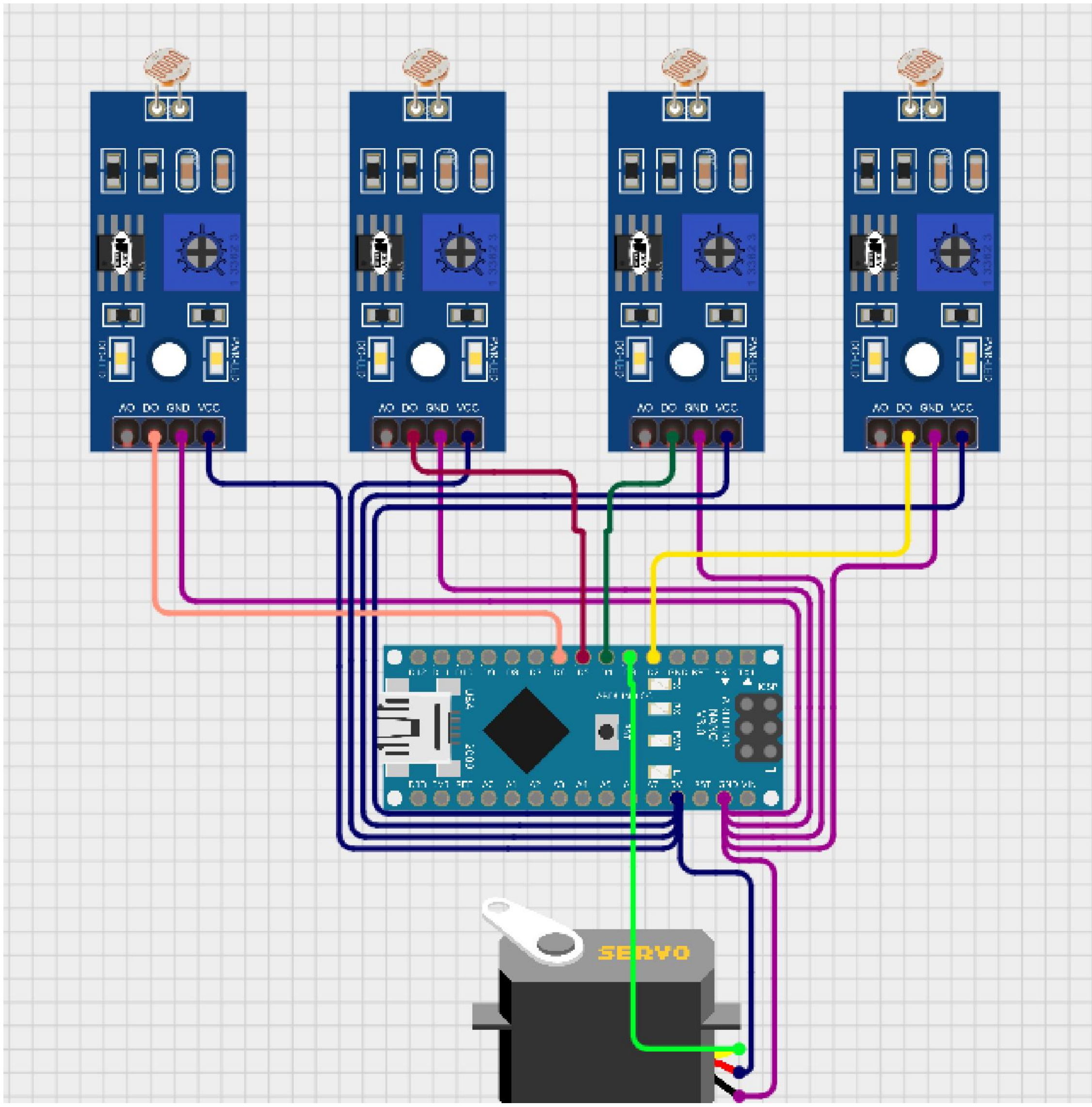
The LDR sensors are mounted around the solar panel to detect light from different directions. Based on the values read, the servo motor adjusts the panel position.

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429 Volume 5, Issue 6, April 2025 Impact Factor: 7.67



V. WORKING PRINCIPLE

The working principle of the smart solar tracker revolves around light intensity detection and directional movement using feedback control. The primary components enabling this functionality are Light Dependent Resistors (LDRs), an Arduino Nano microcontroller, and a servo motor.

LDR Configuration: Four LDRs are arranged in a cross or quadrant pattern—top-left (LDR1), top-right (LDR2), bottom-left (LDR3), and bottom-right (LDR4). This arrangement allows the system to detect the direction of the most intense sunlight. When sunlight falls unevenly on the sensors, the differences in light intensity between the LDRs help determine the sun's position in the sky.

Microcontroller Operation: The LDRs are connected to digital pins on the Arduino Nano. The Arduino reads the digital signals (HIGH or LOW) from each LDR. A HIGH signal indicates the presence of strong light, while a LOW signal indicates weaker or no light.

The Arduino's firmware contains logic to compare the readings from opposite LDRs:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, April 2025
Impact Factor: 7.67

If LDR1 (left top) detects higher light than LDR3 (left bottom), the Arduino increases the servo angle to rotate the panel toward the brighter side.
If LDR3 is brighter than LDR1, the servo angle is reduced to shift the panel back.
Similarly, LDR2 and LDR4 are compared for right-side detection and balance.
This logical comparison ensures that the panel slowly follows the sun's movement across the sky, adjusting its angle every few seconds for optimal alignment.
Servo Motor Action: The servo motor is controlled by the Arduino through PWM signals. As the Arduino updates the desired position (servoPos variable), the servo rotates to the corresponding angle, physically moving the attached solar panel.
Feedback Loop: This process is repeated continuously in the Arduino's loop() function. At regular intervals (e.g., every 200 ms), the system reads sensor values, decides the new servo angle based on light comparison, and adjusts the servo motor's position. This real-time feedback mechanism allows the solar panel to stay aligned with the sun throughout the day.
Power Efficiency: Because the system uses digital signals and a lightweight servo, it consumes minimal power—an advantage for autonomous solar systems. Additionally, since the servo is only actuated when there is a change in light direction, unnecessary movement is avoided, improving energy conservation.
In summary, the smart solar tracker effectively mimics the behavior of a sunflower, turning itself toward the sun using real-time sensor data and microcontroller-based control. This enhances the solar panel's efficiency without the need for manual adjustments or expensive commercial tracking systems.

VI. ARDUINO CODE OVERVIEW:

The heart of the solar tracker system lies in its embedded software, written in Arduino's C-based programming language. The code is responsible for interfacing with the LDR sensors, making decisions based on light intensity, and controlling the servo motor to align the solar panel. Below is a detailed breakdown of the code functionality and logic.

6.1 Libraries and Definitions

```
#include <Servo.h>
```

The Servo.h library is included to simplify controlling the servo motor using PWM (Pulse Width Modulation).
This allows precise control over the servo angle (from 0 to 180 degrees) using the write() function.

```
#define LDR1 2
#define LDR2 4
#define LDR3 5
#define LDR4 6
```

These define the pin numbers for the four LDRs connected to digital inputs on the Arduino Nano.
Each LDR is placed in a specific quadrant to detect light intensity from different directions.

6.2 Initialization

```
Servo myServo;
intservoPos = 90;
```

An object myServo is declared from the Servo class.
servoPos holds the current position of the servo. It is initialized to 90°, which is the center position, simulating a neutral solar panel orientation.

```
void setup() {
Serial.begin(115200);
pinMode(LDR1, INPUT);
pinMode(LDR2, INPUT);
pinMode(LDR3, INPUT);
pinMode(LDR4, INPUT);
```

```
myServo.attach(3);
myServo.write(servoPos);
}
```

Serial.begin(115200) initializes serial communication to display real-time LDR readings and servo positions for debugging.

The four LDR pins are set as input.

The servo is attached to digital pin 3 and moved to the center position at startup.

6.3 Main Loop Logic

```
void loop() {
bool ldr1 = digitalRead(LDR1);
bool ldr2 = digitalRead(LDR2);
bool ldr3 = digitalRead(LDR3);
bool ldr4 = digitalRead(LDR4);
```

Reads digital values (HIGH or LOW) from each LDR.

HIGH usually means more light, and LOW indicates shadow or less light, depending on the threshold voltage.

```
Serial.print("LDR1: "); Serial.print(ldr1);
Serial.print(" | LDR2: "); Serial.print(ldr2);
Serial.print(" | LDR3: "); Serial.print(ldr3);
Serial.print(" | LDR4: "); Serial.println(ldr4);
```

These lines display the LDR status in the Serial Monitor, which is helpful for testing and observing behavior in real time.

6.4 Servo Movement Conditions

```
if (ldr1 == HIGH && ldr3 == LOW &&servoPos< 180) {
servoPos += 5;
 }
 else if (ldr3 == HIGH && ldr1 == LOW &&servoPos> 90) {
servoPos -= 5;
 }
 if (ldr2 == HIGH && ldr4 == LOW &&servoPos> 0) {
servoPos -= 5;
 }
 else if (ldr4 == HIGH && ldr2 == LOW &&servoPos< 90) {
servoPos += 5;
 }
```

These conditional checks compare diagonally opposite LDRs:

LDR1 vs LDR3: Controls right-side tracking.

LDR2 vs LDR4: Controls left-side tracking.

The servo position (servoPos) is incremented or decremented in steps of 5 degrees to gently move the solar panel.

The code includes limits (0 to 180 degrees) to prevent over-rotation of the servo.

6.5 Servo Update and Delay

```
myServo.write(servoPos);
Serial.print(" Servo Position: "); Serial.println(servoPos);
delay(200);
}
```

Updates the servo with the new position.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, April 2025
Impact Factor: 7.67

Displays the updated position in the Serial Monitor.
A short delay (200 milliseconds) is added to avoid rapid or jittery movements, allowing the servo to complete its rotation before reading new sensor data.

6.6 Summary of Code Functionality

The code reads real-time light levels using LDRs.
Based on light comparison, the panel moves slightly left or right using a servo.
Smooth tracking is achieved using incremental movement logic.
Feedback is displayed via the serial monitor.
The code is lightweight, responsive, and runs autonomously, making it ideal for continuous sun tracking throughout the day.
This user manual provides step-by-step guidance on setting up, operating, and troubleshooting the Smart Solar Tracker system built with an Arduino Nano and LDR sensors. It is designed for students, hobbyists, and project developers who want to replicate or understand the system's working.

VII. USER MANUAL

7.1 Hardware Setup Instructions

Components Required:
Arduino Nano (1x)
Light Dependent Resistors (LDR) – Digital Modules (4x)
Servo Motor (SG90 or MG90S recommended) (1x)
Solar Panel (lightweight mini panel for demo or small setup)
Jumper Wires
Breadboard or Nano Expansion Board
Power Supply (USB or 5V regulated)
Mounting structure for panel and sensors
Connection Guide:
Connect the four LDRs to digital pins D2, D4, D5, and D6 of the Arduino Nano.
Attach the servo motor signal wire to digital pin D3 of the Arduino Nano.
Power the servo using the 5V and GND pins of the Arduino (external power may be required for full-scale operation).
Secure the solar panel to the servo shaft or servo arm.
Mount the LDRs in a cross formation around the panel (top-left, top-right, bottom-left, bottom-right).
Powering the System:
Power the Arduino Nano through USB (for programming and testing).
Use a 5V external power source or battery if running the system standalone (ensure servo has enough current capacity).

7.2 Software Setup Instructions

Required Software:
Arduino IDE (install from https://www.arduino.cc/en/software)
USB drivers for Arduino Nano (CH340 or ATmega328P drivers if required)
Programming Steps:
Open the Arduino IDE.
Connect the Arduino Nano via USB.
Select the correct board and port:
Board: Arduino Nano
Processor: ATmega328P (Old Bootloader if needed)
Port: COMx (as detected by your system)
Upload the provided code (from your document) to the board.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, April 2025
Impact Factor: 7.67

Open Serial Monitor at 115200 baud rate to see real-time LDR readings and servo positions.

7.3 How to Operate the System

Place the tracker system under direct sunlight or an artificial light source.
Ensure all four LDRs are positioned correctly and not blocked.
Turn on the power.
Observe the system:
The servo should gradually move the solar panel towards the side with the most light.
When light moves (simulating sun movement), the panel should follow slowly.
Use the Serial Monitor to debug or track sensor performance.

7.4 Calibration and Tips

Initial Positioning: Place the panel in a horizontal position (facing up) with servo angle initialized at 90°.
Light Simulation: During testing, you can simulate sunlight using a flashlight and observe the panel's movement.
Speed Tuning: Adjust the delay(200); in the code for faster or slower movement.
Servo Limits: Ensure your servo doesn't exceed physical movement boundaries. The software limits it between 0° to 180°.
Power Consideration: If the servo behaves erratically, use an external 5V power supply for the servo.

7.5 Troubleshooting

Issue	Cause	Solution
Servo not moving	Improper power or wiring	Check connections and ensure 5V supply is adequate
No LDR readings	Faulty LDR module or wrong pins	Check wiring and replace faulty LDRs
Servo jitters constantly	Noise in sensor data	Increase delay or add hysteresis in code
No COM port detected	Driver issue	Reinstall USB driver or check USB cable

7.6 Safety and Maintenance

Do not expose electronics to moisture or rain.
Avoid overloading the servo motor; use only small lightweight solar panels.
Disconnect power when making changes to the wiring.

VIII. APPLICATIONS

- Renewable energy research and education
- Rural solar energy projects
- Low-power IoT solar systems
- Portable solar charging kits

IX. FUTURE SCOPE

- Upgrade to dual-axis tracking for vertical movement.
- Add analog LDRs for more precise light measurement.
- Include solar power monitoring (voltage/current sensors).
- Use wireless modules (e.g., ESP32) for remote monitoring.

X. CONCLUSION

The smart solar tracker using Arduino Nano and LDR sensors offers a simple, affordable, and efficient solution to enhance solar energy harvesting. By automating the tracking process, it ensures optimal panel orientation, contributing to higher energy output. This project demonstrates how embedded systems and renewable energy technologies can be combined to address global energy challenges effectively.

REFERENCES

[1]. Arduino Official Documentation (https://www.arduino.cc)
[2]. Research articles on solar tracking efficiency
[3]. Tutorials and community forums for Arduino projects